

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-298072

(43)Date of publication of application : 11.10.2002

(51)Int.Cl.

G06F 19/00  
G06F 12/00

(21)Application number : 2001-101476

(71)Applicant : FUJITSU LTD

(22)Date of filing : 30.03.2001

(72)Inventor :  
TERAI YOSHIKI  
KAWAMURA ISAMU  
AMAHISA HIDEYUKI

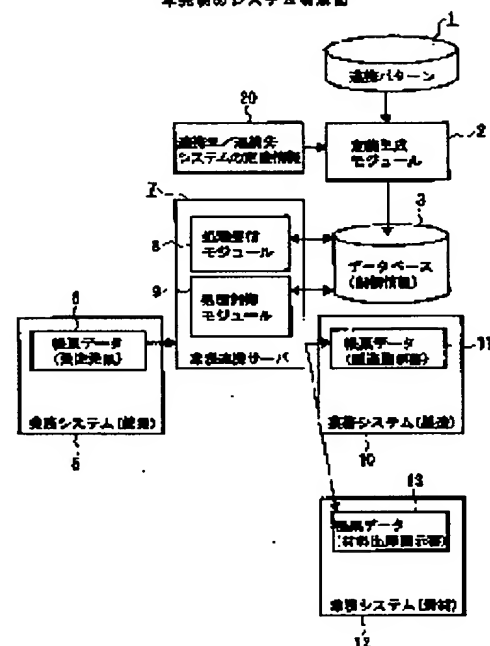
## (54) TIE-UP METHOD AND PROGRAM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a system automatically executing a series of processing of system tie-up, easily adding and correcting functions, and quickly processing the data according to emergence by only predefining definition information of the data of the respective systems.

**SOLUTION:** This tie-up method and program is provided with a step selecting an applicable pattern from interlocking patterns based on the definition information of the data of the interlocking system and the tie-up counter-systems, a step generating control information setting a plurality of processing according to the selected pattern and an order for executing the plurality of processing, and a step outputting the data generated from the tie-up system to the tie-up counter-systems by executing it according to the control information.

本発明のシステム構成図



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(11)特許出願公開番号  
特開2002-298072  
(P2002-298072A)

(43)公開日 平成14年10月11日(2002.10.11)

(51)Int.Cl. <sup>7</sup>		識別記号	F I	テーマコード* (参考)
G 0 6 F	19/00	3 0 0	G 0 6 F 19/00	3 0 0 N 5 B 0 8 2
	12/00	5 1 1	12/00	5 1 1 C

審査請求 未請求 請求項の数 5 O.L (全 13 頁)

(21)出願番号 特願2001-101476(P2001-101476)

(22)出題日 平成13年3月30日(2001.3.30)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 發明者 寺井 義起

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(72)発明者 河村 勇

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74) 代理人 100089141

弁理士 岡田 守弘

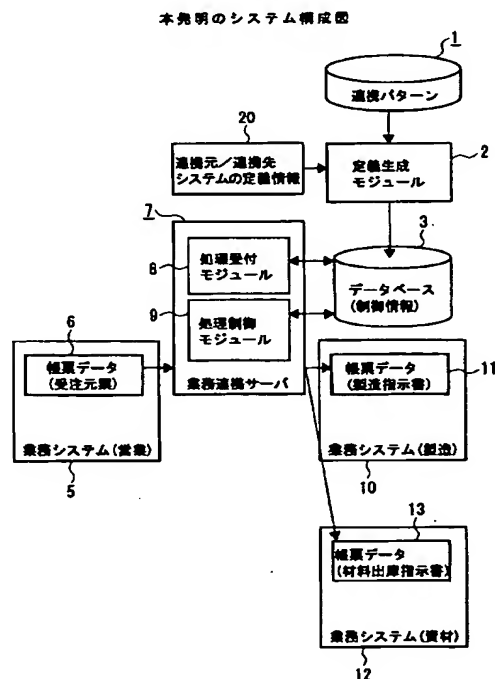
**最終頁に続く**

(54) 【発明の名称】 連機方法および連機プログラム

(57) 【要約】 (修正有)

【課題】 各システムのデータの定義情報を予め定義するのみでシステム間連携の一連の処理を自動実行、機能の追加修正を簡易に実現、および緊急性に応じて迅速にデータ処理するシステムを実現することを目的とする。

【解決手段】 連携元および連携先のシステムのデータの定義情報をもとに、連携パターン中から該当パターンを選択するステップと、選択されたパターンに従い複数の処理および該複数の処理を実行する順番を設定した制御情報を生成するステップと、連携元システムからのデータについて、該当制御情報に従い実行することで生成されたデータを連携先のシステムに出力するステップとを有する連携方法および連携プログラムである。



## 【特許請求の範囲】

【請求項 1】 システム間でデータの連携を行う処理連携方法において、

連携元および連携先のシステムのデータの定義情報をもとに、連携パターン中から該当パターンを選択するステップと、

上記選択されたパターンに従い複数の処理および該複数の処理を実行する順番を設定した制御情報を生成するステップと、

連携元システムからのデータについて、該当制御情報に従い実行することで生成されたデータを連携先のシステムに出力するステップとを有する連携方法。

【請求項 2】 上記制御情報を即実行可能な文書構造に変換するステップを有する請求項 1 記載の連携処理方法。

【請求項 3】 システム間でデータの連携を行う連携プログラムにおいて、

連携元および連携先のシステムのデータの定義情報をもとに、連携パターン中から該当パターンを選択する手段と、

上記選択されたパターンに従い複数の処理および該複数の処理を実行する順番を設定した制御情報を生成する手段と連携元システムからのデータについて、該当制御情報に従い実行することで生成されたデータを連携先のシステムに出力する手段として実行させるための連携プログラム。

【請求項 4】 上記制御情報を即実行可能な文書構造に変換する手段を有する請求項 3 記載の連携プログラム。

【請求項 5】 上記制御情報をもとに格納媒体、移動契機、および帳票データ単位の 1 つ以上に予め付与されている優先度をもとに当該制御情報の優先レベルを決定して当該優先レベルに対応づけて格納する手段を有する請求項 4 記載の連携プログラム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、システム間でデータの連携を行う連携方法および連携プログラムに関するものである。企業内の異なる業務システム間や企業間のシステムを、ハブ機能を設けて相互に連携したり、データやプロセスを統合したりするアプローチを企業アプリケーション統合(Enterprise Application Integration)といい、業務システムのすみやかな統合と柔軟な変更を可能とし、ビジネス業務の自動化や既存システムの資産の有効活用する手法として注目されている。

【0002】 企業アプリケーション統合を実現するためには、業務システム間の電子帳票データの交換処理に必要な入出力(集配信を含む)、フォーマット変換、コード変換、単票の一括形式化、一括形式の単票化、振り分けなどのいろいろな機能要素が必要となる。更に、業務システムにはいろいろな運用形態があり、構築および運用は非常に煩雑となっている。

## 【0003】

【従来の技術】 従来、企業内、企業間、および企業と消費者間の業務システムを構築するためには、入出力(集配信を含む)、フォーマット変換、コード変換、単票の一括形式化、一括形式の単票化、振り分けなどのいろいろな機能や、これらの組み合わせおよび手順を個別にアプリケーションなどで構築することで実現していた。その際、業務の実態に応じてそれぞれの処理の内容とその順番を意識するようにしていた。

## 【0004】

【発明が解決しようとする課題】 上述したように企業内、企業間および企業と消費者間の業務システムを連携する場合に、業務システムが流通する帳票データに対して、入出力(集配信を含む)、フォーマット変換、コード変換、単票の一括形式化、一括形式の単票化、振り分けなどの処理が必要となり、業務の実態に応じてそれぞれの処理を構成および順序つける必要がある。従来は、個別のアプリケーション開発や設定作業が必要となってしまうと共に、処理が煩雑化してしまうという問題があった。

【0005】 また、入出力などの処理の定義情報をもとに実行時に実行するために、アプリやプラットフォームに依存した形式に編集する必要があると共に、機能追加などのときに初めから編集をやり直すなどの手間が必要となってしまう問題もあった。

【0006】 また、業務システム間でデータ連携する場合、バッチ処理的な緊急性を要しない場合や、リアルタイム処理のような緊急性を要する場合があり、これらに柔軟に対処できないという問題もあった。

【0007】 本発明は、これらの問題を解決するため、連携しようとするシステムのデータの定義情報をもとにシナリオから該当するものを選択して最適な処理とその順番を自動決定して実行したり、制御情報を即実行可能形式の文書構造に変換したり、制御情報の優先レベルを決定して優先レベル別に格納してデータを処理したりし、各システムの定義情報を予め定義するのみでシステム間連携の一連の処理を自動実行、機能の追加修正を簡易に実現、および緊急性に応じて迅速に処理するシステムを実現することを目的としている。

## 【0008】

【課題を解決するための手段】 図 1 を参照して課題を解決するための手段を説明する。図 1 において、定義生成モジュール 2 は、連携元および連携先のシステムの定義情報をもとに、該当連携パターンから選択したパターンに従い処理および順番を設定した制御情報を生成したりなどするものである。

【0009】 処理制御モジュール 9 は、連携元システムから入力されたデータについて該当制御情報に従い所定の処理を所定の順番で実行して生成したデータを連携先のシステムに出力したりなどするものである。

【0010】次に、動作を説明する。定義生成モジュール2が連携元および連携先のシステムの定義情報をもとに連携パターンの中から選択されたパターンに従い処理およびその順番を設定した制御情報を生成し、処理制御モジュール9が連携元システムから受信したデータについて該当制御情報に従い所定の処理を所定の順番で実行して生成したデータを連携先のシステムに出力するようにしている。

【0011】また、定義生成モジュール2は、制御情報を即実行可能なXML文書などの文書構造を生成するようにしている。また、処理制御モジュール9は、制御情報をもとに格納媒体、移動契機、および帳票データ単位の1つ以上に付与されている優先度をもとに当該制御情報の優先レベルを決定して当該優先レベルに対応づけて格納してデータ処理を行い、連携先に出力するようにしている。

【0012】従って、連携しようとするシステムのデータの定義情報をもとにシナリオから該当するものを選択して最適な処理と順番を自動決定して実行したり、制御情報を即実行可能形式の文書構造に変換したり、制御情報に優先レベルを決定して優先レベル別に格納してデータ処理したりすることにより、各システムの定義情報を予め定義するのみでシステム間連携の一連の処理を自動実行、機能の追加修正を簡易に実現、および緊急性に応じて迅速に処理するシステムを実現することが可能となる。

#### 【0013】

【発明の実施の形態】次に、図1から図12を用いて本発明の実施の形態および動作を順次詳細に説明する。

【0014】図1は、本発明のシステム構成図を示す。図1において、連携パターン1は、連携元および連携先の定義情報の内容で決まるパターンを登録したものである（図6参照）。

【0015】定義生成モジュール2は、連携元、連携先のシステムの定義情報20をもとに、連携パターン1の中から該当するパターンを選択し、選択したパターンに従い所定の処理を所定の順番で設定した制御情報（定義情報）を生成したり、下記の優先度を予め定義したりなどするものである。

#### 【0016】優先度：

- ・データ格納媒体：メッセージキュー>>ワークフロー>データベース=ファイル
- ・起動契機：API起動>ワークフロー起票>コマンド=集信監視など
- ・帳票データ単位：単票>>一括形式

連携元／連携先システムの定義情報20は、連携元のシステムおよび連携先のシステムの帳票データなどの定義情報である。

【0017】データベース3は、定義生成モジュール2によって生成された連携元と連携先のシステムに対応づ

けた定義情報（制御情報）を格納したものであって、連携元から受信したデータ（例えば帳票データ）を連携先の帳票データに変換するための定義情報を格納したものである。

【0018】業務システム（営業）5は、各種業務を行うシステムの例であって、ここでは、営業システムであり、受注元票を業務連携サーバ7に送信する連携元となるシステムである。

【0019】帳票データ6は、業務システム5が業務連携サーバ7に送信する連携元の帳票データ（受注元票）である。業務連携サーバ7は、連携元のシステムから受信したデータについて、データベース3に登録されている該当定義情報（制御情報）に従い、連携先の定義情報に合わせて変換し、連携先のシステムに送信したりなどするものであって、ここでは、処理受付モジュール8および処理制御モジュール9などから構成されるものである。

【0020】処理受付モジュール8は、連携元のシステム（例えば業務システム（営業）5から帳票データ6を受け付けたり（受信したり）などするものである。処理受付モジュール8は、処理要求を受け付け、通常、帳票データを入力するのは処理制御モジュールである。処理開始要求と同時に帳票データも与えられるインタフェースもある。

【0021】処理制御モジュール9は、処理受付モジュール8で受け付けたデータ（例えば帳票データ）について、連携元と連携先のシステムに対応づけてデータベース3に登録されている定義情報（制御情報）に従い、受け付けたデータを変換して連携先のシステムに送信したりなどするものである。

【0022】業務システム（製造）10および業務システム（資材）12は、連携先のシステムの例であって、業務連携サーバ7で変換された後のデータを受信し、業務連携を行うシステムである。業務システム（製造）10では、帳票データ（製造指示書）11を受信する。業務システム（資材）12では、帳票データ（材料出庫指示書）13を受信する。

【0023】次に、図2のフローチャートの順番に従い、図1の構成の動作を順次詳細に説明する。図2は、本発明の動作説明フローチャートを示す。

【0024】図2において、S1は、各システムの定義を行う。これは、例えば各システムの帳票の定義情報をそれぞれ定義し、図1の業務連携サーバ7に登録する。S2は、連携システムの指定を行う。これは、例えばS1で定義して業務連携サーバ7に登録したシステムの定義情報から、連携元のシステムおよび連携先のシステムの指定を行う。

【0025】S3は、解析する。これは、S2で指定された連携元のシステムおよび連携先のシステムの、S1で定義されたそれぞれの定義情報を読み出し、これら連

携元のシステムの定義情報および連携先のシステムの定義情報について解析する。

【0026】S4は、シナリオを判断する。これは、S3で解析した結果をもとに、連携元のシステムの定義情報および連携先のシステムの定義情報に該当するパターンを、後述する図6の連携パターン（パターン1, 2, 3, 4, 5・・・）の中から該当パターンを判断する。

【0027】S5は、連携パターンの中から該当パターンを選択する。S6は、順番を決定する。これは、S5で選択した該当パターンに従い、処理の順番を決定する。例えば図6のパターン1の場合には、当該パターンに従い、データ入力、集約、一括変換、単票分解、データ出力という処理の順番を決定する。

【0028】S7は、制御情報として設定する。これは、S6で決定された処理の順番を制御情報（連携元のシステムから連携先のシステムに帳票データを渡すときに変換する定義情報）として設定、例えば後述する図5の上段に示す制御情報として設定する。

【0029】以上の手順によって、予め各システムの帳票データの定義情報を定義して図1の業務連携サーバ7に登録しておき、必要に応じて連携元のシステムと連携先のシステムとを指定する（S2）のみで自動的に連携元のシステムの帳票データから連携先のシステムの帳票データに変換する制御情報（連携元システムから連携先システムへの帳票の変換の定義情報）を自動作成して設定することが可能となる。以下図3から図6を用いて詳細に説明する。

【0030】図3は、本発明の1実施例構成図を示す。図3において、連携元業務システムの定義情報21は、連携元のシステムの定義情報であって、例えば既述した図1の業務システム（営業）5の帳票データ（受注元票）6の定義情報である。

【0031】連携先業務システムの定義情報22は、連携先のシステムの定義情報であって、例えば既述した図1の業務システム（製造）10の帳票データ（製造指示書）である。

【0032】シナリオ選択・制御情報生成モジュール23は、連携元業務システムの定義情報21から連携先業務システムの定義情報22に帳票データを変換する該当パターンを、後述する図6の連携パターン（シナリオ25）の中から選択したり、選択した該当パターンに従い処理の順番を決定して制御情報26として設定したりなどするものである。

【0033】データベース24は、シナリオ25および制御情報26を登録して管理するものである。シナリオ25は、連携元のシステムの定義情報および連携先のシステムの定義情報から生じる組合わせのパターン（処理（例えば図6に示すように、データ入力、集約、一括変換、単票分解、データ出力、変換、振り分け、単票分割などの処理）とその順番）を予め登録したものである。

【0034】制御情報26は、連携元のシステムおよび連携先のシステムが指定されたときに連携元のシステムの帳票データを連携先のシステムの帳票データに変換する制御情報（定義情報）を登録したものである（既述した図2のフローチャートに従い自動作成して登録したものである）。

【0035】実行制御モジュール27は、帳票データ28を受信したときに、連携元のシステムおよび連携先のシステムに対応づけて登録されている当該制御情報26を読み出して当該制御情報26に定義（設定）されている処理およびその順番に従い変換した帳票データ29を連携先のシステムに出力したりなどするものである。

【0036】次に、動作を説明する。

（1）シナリオ選択・制御情報生成モジュール23が指定された連携元業務システムの定義情報21および連携先業務システムの定義情報22をもとに、データベース24内のシナリオ25の中から該当パターンを選択する。例えば後述する図6の連携パターン（シナリオ25）の中から該当する1つのパターンを選択する。

（2）（1）で選択したパターンに従い、処理とその順番を決定して制御情報に設定し、データベース24に登録する。

（3）連携元のシステムから帳票データ28を受信したときに、連携元のシステムおよび連携先のシステムに該当する制御情報26をデータベース24から読み出し、当該読み出した制御情報26に従い、所定の処理を所定の順番に実行し、連携元のシステムの帳票データ28から連携先のシステムの帳票データ29に自動的に変換し、連携先のシステムに出力する。

【0038】以上のように、連携元のシステムおよび連携先のシステムを指定するのみで、それぞれの定義情報を読み込んで図6の連携パターン（シナリオ25）の中から該当パターンを選択し、選択されたパターンに従い処理とその順番を制御情報26として自動設定し、実行時に連携元のシステムから受信した帳票データ28を該当制御情報26に従い連携先のシステムの帳票データ29に自動変換して出力することが可能となる。

【0039】図4は、本発明の説明図を示す。図4の（a）、（b）は、先行業務（連携元業務）と後続業務（連携先業務）の定義情報に従い、いずれのシナリオを適用するかを決定するものである。

【0040】ここで、シナリオの内容は、アクション（業務システムの結びつき、あるいは組合わせ）定義で選択した論理ノード（業務システムであって、ここでは先行業務システム）のグループ化の有無（図1では後続業務が業務システム（製造）10および業務システム（資材）12の2つをグループ化するので、この場合には後続業務にグループ化有りとなる）や、入出力論理ノードが扱う伝票のフォーマット、さらに、出力論理ノードの伝票形式により決定する。業務連携定義では、利用

者により、アクション定義で入力論理ノード（または入力論理ノードグループ）と出力論理ノードグループの組み合わせが指定された時点で、シナリオとして最適な処理内容をいくつか選択し、その内容を利用者に通知する。利用者は、業務連携定義情報が通知されたシナリオの中から業務にあった内容を決定する。

【0041】尚、図4の表に示すように組み合わせによりエラーとなる場合がある。本エラーは、アクション定義において、入力論理ノード（または入力論理ノードグループ）と出力論理ノード（または出力論理ノードグループ）の組み合わせを選択した時点で通知する。

【0042】ここで、図4の表中の縦軸の先行業務および横軸の後続業務内の帳票データのフォーマット（C I I, EDIFACT, ユーザ固有形式）、帳票単位（単票、一括形式など）である。これらの組み合わせのときに、図示のようにシナリオ1, 2・・・などあるいはエラーと判定する。エラーの場合には、変換処理を行わない。

【0043】図5は、本発明の制御情報と帳票データの例を示す。図5中で上段が制御情報であり、下段が帳票データの例を示す。ここで、制御情報の最上段のシナリオ（①入力、②振分、③変換、④分割、⑤出力）は、後述する図6の最下段のパターン4に従い制御情報を設定したものであり、これら①から⑤の順番に処理を実行する様子を示す（矢印でその処理の一部を示す）。尚、図4の制御情報および帳票データを、XML形式で分かり易く後述する図9に記載する。

【0044】図6は、本発明の連携パターン例を示す。これは、既述した図3のシナリオ25に対応するものであって、ここでは、入力（単票、一括）、出力（単票、一括）、出力先（1箇所、複数箇所）の全組み合わせのうち帳票データの連携処理を必要のある場合の例を4つ示す。ここでは、上段から図示のような組み合わせの場合のパターンである。

【0045】

入力	出力	出力先
・パターン1：単票又は一括	単票	1箇所
・パターン2：単票又は一括	単票	複数箇所
・パターン3：単票又は一括	一括	1箇所
・パターン4：単票又は一括	一括	複数箇所
・・・		

以上のように、連携元のシステムの帳票データの定義情報および連携先のシステムの帳票データの定義情報をもとに、図1の業務連携サーバ7で処理を行う内容（入力、出力、出力先）に一致するパターンを選択（既述した図2のS4からS5）し、選択したパターンの処理（例えばパターン4のデータ入力、振分／集約、変換、データ出力）とその順番に従い制御情報26を設定することが可能となる（図5）。

【0046】図7は、本発明の動作説明フローチャート（文書構造化）を示す。図7において、S11は、定義

情報を読み込む。これは、各システムの例えば帳票データの定義情報を読み込む。

【0047】S12は、シナリオ情報を解析する。これは、文書構造化しようとする連携元のシステムの帳票データの定義情報から連携先のシステムの帳票データの定義情報に変換するときに合致するシナリオ情報（例えば既述した図6の連携パターン中から選択したパターン4）を解析する。

【0048】S13は、処理を順番に書き出す。例えばS12で既述した図6のパターン4を選択したときは、処理として、データ入力、振分／集約、変換、データ出力を順番に書き出す。

【0049】S14は、入力情報を解析する。これは、S12で解析したシナリオの入力情報を解析し、ここでは、入力情報を抽出する。尚、S20で使う出力情報、フォーマット情報（入力、出力）なども合わせて抽出して保持する。

【0050】S15は、先頭の処理を取り出す。ここでは、例えば既述した図6のパターン4の場合には、先頭の「データ入力」という処理を取り出す。以降同様に、次のループ時に、振分／集約、変換、データ出力という順に処理を取り出す。

【0051】S16は、入力ノードを検索する。これは、S15で取り出した処理、例えばデータ入力処理における入力ノードを、innode000001として取り出す（図9の①入力定義参照）。

【0052】S17は、入力情報をもとにXML形式に書き出す。これにより、後述する図9の例えば①入力定義と、その入力ノードinnode000001がXML形式で書き出されることとなる。

【0053】S18は、入力終わりが判別する。YESの場合には、S19に進む。NOの場合には、次の入力について、S16以降を繰り返して入力ノードをXML形式で順番に書き出す。

【0054】S19は、処理終わりが判別する。これは、S13で順番に書き出した処理（例えばデータ入力、振分／集約、変換、データ出力）の全てについて終了したか判別する。YESの場合には、S20に進む。NOの場合には、S15以降を繰り返し次の処理について実行する。

【0055】S20は、S15からS19のYESによって、入力情報について全ての処理（例えばデータ入力、振分／集約、変換、データ出力）について全て終了したので、次に、出力情報についてもS15からS19のYESと全く同様に各処理の出力ノードを検索してXML形式として書き出す。

【0056】S21は、完成する。以上の手順によって、連携元のシステムの帳票データの定義情報から連携先のシステムの帳票データの定義情報に変換するシナリオ（S12、図6の例えばパターン4）を解析し、当該

シナリオ中の各処理の入力情報、出力情報をそれぞれXML形式で書き出し、完成したこととなる（図9の上段の部分）。

【0057】図8は、本発明の動作説明フローチャート（文書構造化、その2）を示す。ここで、例えば同じ“交換”であっても、細かく言えば他に分割、集約があり、また、シナリオによっては展開パターンが異なることがあるので、その一例をもとに以下説明する。

【0058】図8において、S32は、入力か判別する。これは、既述した図7でシナリオから順番に取り出したものが、入力か判別する。YESの場合には、S32からS34を実行する。NOの場合には、S32からS34をスキップしてS35に進む。

【0059】S32は、S31のYESで入力と判別したので、先頭の処理を取り出す。S33は、S32で取り出した処理について、入力ノードの数だけ検索し、ラベルを取り出す。例えばラベルinnode000001を取り出す。

【0060】S34は、展開する。これにより、入力ノードの数だけ検索したラベルがXML形式で展開されることとなる。以上のS31のYES、S32からS34によって、後述する図9の制御情報の中段の①入力のフィールドに、入力ノードinnode000001が図示のようにXML形式で自動的に展開されることとなる。

【0061】S35は、振分か判別する。YESの場合には、S36に進む。NOの場合には、スキップしてS39に進む。S36は、入力ノードを検索し、ラベルを取り出す。

【0062】S37は、出力ノードの数だけ検索し、ラベルを取り出す。S38は、展開する。以上のS35のYES、S36からS38によって、後述する図9の制御情報の中段の②振分のフィールドに、入力ノードinnode000001に対応づけて出力ノードoutnode000001、outnode000002、outnode000003の3つが図示のようにXML形式で自動的に展開されることとなる。

【0063】S39は、変換あるいは出力か判別する。YESの場合にはMS40に進む。NOの場合には、スキップして終了する。S40は、出力ノードを検索し、ラベルを取り出す。

【0064】S41は、展開する。以上のS39のYES、S40、S41によって、後述する図9の制御情報の中段の③変換（あるいは④出力）のフィールドに、出力ノードoutnode000001、outnode000002、outnode000003の3つが図示のようにXML形式で自動的に展開されることとなる。

【0065】図9は、本発明の制御情報の文書構造例を示す。これは、既述した図6の連携パターン中のパター

ン4について、図7、図8のフローチャートに従い、プラットフォームを選ばない即実行可能形式であるXML形式の文書構造に自動生成したものである。

【0066】図9において、上段と中段は、制御情報である。上段は、既述した図7のフローチャートに従い、例えば図6の連携パターン中のパターン4および定義情報をもとに自動生成したものである。中段は既述した図8のフローチャートに従い上段を即実行可能形式のXML形式に展開した様子を模式的に示したものである。

【0067】下段は、帳票データの例を示す。図10は、本発明の動作説明フローチャート（レベル付け）を示す。これは、定義情報についていずれの優先レベルに対応づけて格納するかを決定するフローチャートである。

【0068】図10において、S41は、データ格納媒体を解析して重みを設定する。これは、定義情報（制御情報）について、データ格納媒体を解析して重みを設定する。例えばデータ格納媒体の種類によって下記の該当する重みを設定する。

【0069】・メッセージキュー：3

・ワークフロー：2

・データベース：1

・ファイル：1

S42は、起動契機を解析して重みを設定する。これは、定義情報について、起動契機を解析して重みを設定する。例えば起動契機の種類によって下記の該当する重みを設定する。

【0070】・API起動：3

・ワークフロー起票：2

・コマンド：1

・集信監視：1

S43は、帳票データ単位を解析して重みを設定する。これは、定義情報について、帳票データ単位を解析して重みを設定する。例えば帳票データ単位の種類によって下記の該当する重みを設定する。

【0071】・単票：3

・一括：1

S44は、重みを集計する。これは、S41からS43で設定した各重みを集計、例えば合計する。

【0072】S45は、集計値が6以上か判別する。YESの場合には、S47で優先レベル1と決定する。NOの場合には、S46に進む。S46は、集計値が3以上か判別する。YESの場合には、S47で優先レベル2と決定する。NOの場合には、優先レベル3と決定する（あるいはレベルを増やしてキャッシュ対象外の例を増加させる）。

【0073】以上によって、定義情報について、データ格納媒体、起動契機、帳票データ単位の種類で重みをそれぞれ設定して集計し、集計値によって優先レベル1、2、3と決定することにより、緊急性の高い帳票データ

の定義情報を優先レベル1に対応づけて格納して迅速に読み出して帳票データを処理することが可能となる。

【0074】図11は、本発明の定義情報制御の処理フローチャートを示す。図11において、S52は、処理要求があったときに開始し、処理要求情報の入力・解析を行う。

【0075】S53は、キャッシュ数上限以下か判別する。YESの場合には、S54に進む。NOの場合には、キャッシュが一杯であるので、S60に進む。S54は、外部記憶装置の定義情報の更新日時を取得する。

【0076】S55は、メモリから定義情報を検索する。検索は、検索木内の優先度大のノードから、参照日時の降順で行う。S56は、S55のメモリ（キャッシュ）検索の結果、ヒットしたか判別する。YESの場合には、S57に進む。NOの場合には、ミスヒットと判明したので、S60に進む。

【0077】S57は、最終参照日時がメモリ上のものと、外部記憶装置上のものとが等しいか判別する。YESの場合には、S58に進む。一方、NOの場合には、メモリ上の定義情報が古いと判明したので、S59でメモリに展開している当該定義情報を削除し、S60に進む。

【0078】S58は、S57のYESでメモリ上の定義情報の更新日時が外部記憶装置上の定義情報の更新日時と等しく、メモリ上の定義情報が最新と判明したので、メモリ上の定義情報を参照し、メモリを参照日時順に順番替えし、終了する（合わせて参照した定義情報をもとに帳票データの処理を実行する）。

【0079】S60は、外部記憶装置から定義情報を参照する。S61は、キャッシュ数上限以下か判別する。YESの場合には、S62に進む。NOの場合には、キャッシュ（メモリ）が一杯であると判明したので、終了する。

【0080】S62は、S61のYESでキャッシュ（メモリ）が一杯でないかと判明したので、定義情報項目を参照する。S63は、定義情報項目（業務システムの通信方式、データ格納媒体、起動契機、帳票データ単位、その他）の優先度を判定し、定義情報の検索優先度を算出する（既述した図10参照）。

【0081】S64は、メモリ格納対象か判別する。これは、S63で算出した優先度（図10で算出した優先レベル）がメモリ（キャッシュ）の格納対象のデータであるか判別する。YESの場合には、S65で定義情報の優先度に応じて、優先度段階に応じたメモリ格納位置に、参照日時順に追加または入替（キャッシュ数上限まで）る。

【0082】以上のように、優先レベルの高い定義情報をメモリに参照日時順に格納することにより、優先レベルの高い定義情報をメモリ上から迅速に参照して処理を実行することが可能となる。

【0083】図12は、本発明のメモリに展開される定義情報の構造例を示す。図12において、格納木のルートノードの配下の左側から順に優先レベル1, 2, 3, ... というように設け、かつ各優先レベル内では参照日時の降順（即ち、参照日時が最近ほど格納木のルートノードに近い方から順に情報（定義情報）を格納する。

【0084】・前リストポインタ：

・次リストポインタ：  
・A定義情報更新日時：  
・B定義情報更新日時：

・・・  
・A定義情報：  
・B定義情報：  
・・・

以上のように、優先レベル毎に参照日時降順に定義情報をリンクし、既述した図11のフローチャートに従い定義情報の参照などを実行することにより、優先レベルの高く、かつ参照日時の新しい程、ヒットしやすかつ検索が簡易にでき、優先レベルの高く、頻繁に参照する定義情報ほど高速に参照して当該定義情報をもとに帳票データの処理実行することが可能となる。

【0085】（付記1）システム間でデータの連携を行う連携方法において、連携元および連携先のシステムの定義情報をもとに、連携パターン中から該当パターンを選択するステップと、上記選択されたパターンに従い複数の処理および該複数の処理を実行する順番を設定した制御情報を生成するステップと連携元システムからのデータについて、該当制御情報に従い実行することで生成されたデータを連携先のシステムに出力するステップとを有する連携方法。

【0086】（付記2）上記制御情報を即実行可能な文書構造に変換するステップを有する付記1記載の連携方法。

【0087】（付記3）上記文書構造をXML文書とした付記1記載の連携方法。

（付記4）定義情報の格納媒体、移動契機、および帳票データの1つ以上に予め付与されている優先度をもとに当該定義情報の優先レベルを決定して当該優先レベルに対応づけて格納するステップを有する付記1から付記3のいずれかに記載の連携方法。

【0088】（付記5）システム間でデータの連携を行う連携プログラムにおいて、連携元および連携先のシステムの定義情報をもとに、連携パターン中から該当パターンを選択する手段と、上記選択されたパターンに従い複数の処理および該複数の処理を実行する順番を設定した制御情報を生成する手段と連携元システムからのデータについて、該当制御情報に従い実行することで生成されたデータを連携先のシステムに出力する手段として実行させるための連携プログラム。

【0089】（付記6）上記制御情報を即実行可能な文



書構造に変換する手段を有する付記 5 記載の連携プログラム。

【0090】（付記 7）定義情報の格納媒体、移動契機および帳票データの 1 つ以上に付与されている優先度をもとに当該定義情報の優先レベルを決定して当該優先レベルに対応づけて格納するステップを有する付記 5 あるいは付記 6 記載の連携プログラム。

【0091】（付記 8）上記制御情報をもとに格納媒体、移動契機、および帳票データ単位の 1 つ以上に予め付与されている優先度をもとに当該制御情報の優先レベルを決定して当該優先レベルに対応づけて格納する手段を有する付記 6 記載の連携プログラム。

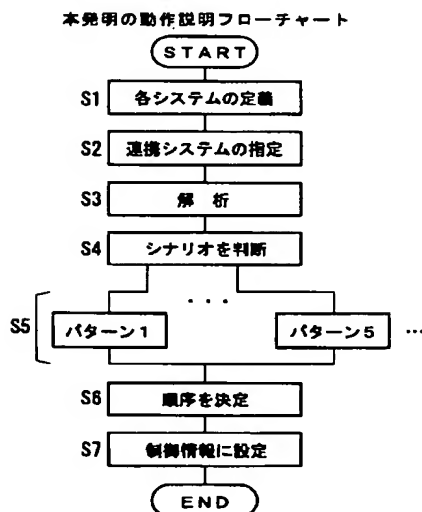
【0092】

【発明の効果】以上説明したように、本発明によれば、連携しようとするシステムのデータの定義情報をもとにシナリオから該当するものを選択して最適な処理とその順番を自動決定して実行したり、制御情報を即実行可能形式の文書構造に変換したり、制御情報のデータ格納媒体などに優先度を付与して当該制御情報の優先レベルを決定して優先レベル別に格納して参照しデータ処理したりする構成を採用しているため、各システムの定義情報を予め定義するのみでシステム間連携の一連の処理を自動実行、機能の追加修正を簡易に実現、および緊急性に応じて迅速にデータ処理するシステムを実現することが可能となる。

【図面の簡単な説明】

【図 1】本発明のシステム構成図である。

【図 2】



【図 2】本発明の動作説明フローチャートである。

【図 3】本発明の 1 実施例構成図である。

【図 4】本発明の説明図である。

【図 5】本発明の制御情報と帳票データの例である。

【図 6】本発明の連携パターン例である。

【図 7】本発明の動作説明フローチャート（文書構造化）である。

【図 8】本発明の動作説明フローチャート（文書構造化、その 2）である。

10 【図 9】本発明の制御情報の文書構造例である。

【図 10】本発明の動作説明フローチャート（レベル付け）である。

【図 11】本発明の定義情報制御の処理フローチャートである。

【図 12】本発明のメモリに展開される定義情報の構造例である。

【符号の説明】

1：連携パターン（シナリオ）

2：定義生成モジュール

20 3：データベース（制御情報）

5、10、12：業務システム

7：業務連携サーバ

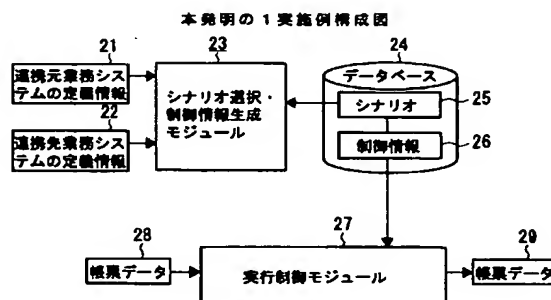
8：処理受付モジュール

9：処理制御モジュール

11、13：帳票データ

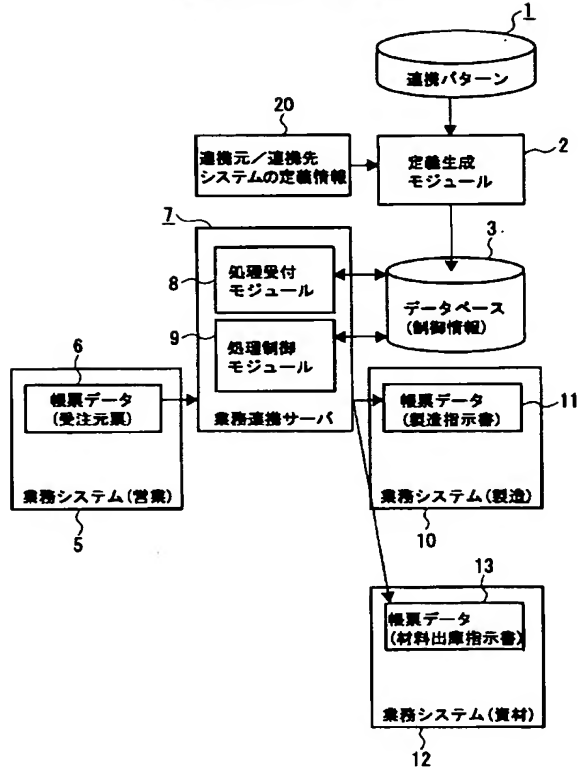
20：連携元／連携先システムの定義情報

【図 3】



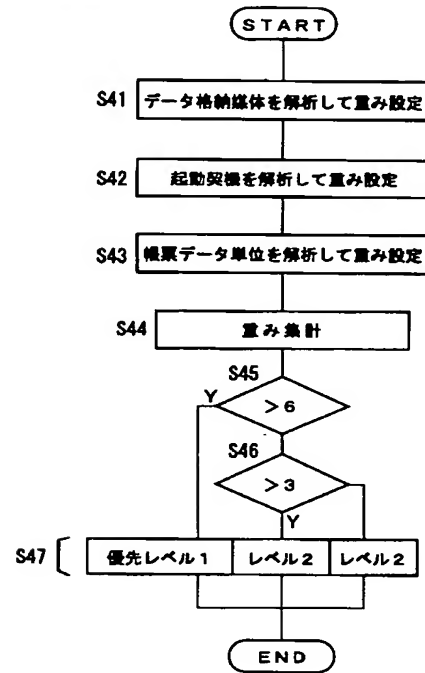
【図 1】

本発明のシステム構成図



【図 10】

本発明の動作説明フローチャート(レベル付け)



【図 4】

本発明の説明図

(a)

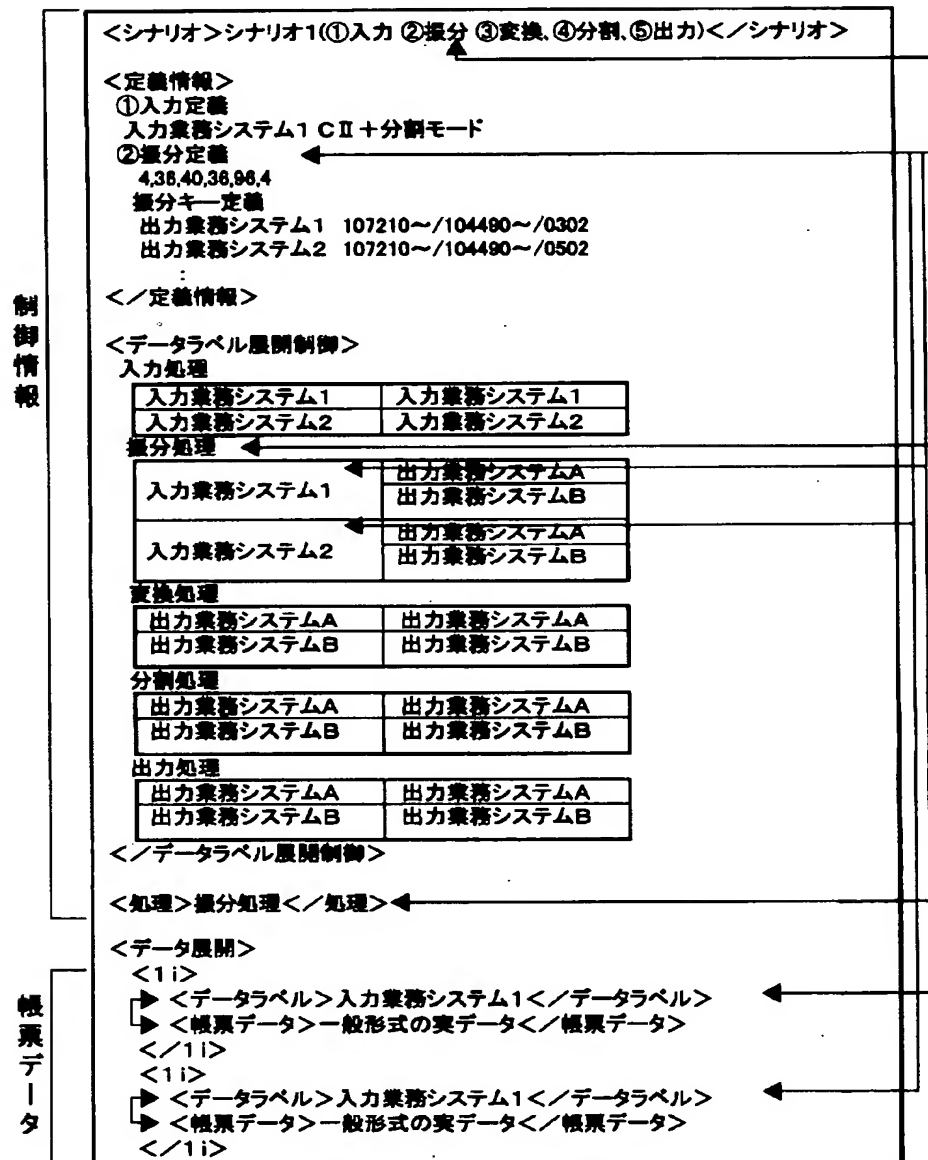
後続業務		出力論理ノード(伝票形式: 単票)				出力論理ノード(伝票形式: 一括)		
		ECOFAC	CII	ユーザ固有フォーマット		CII	ユーザ固有フォーマット	
先行業務	入力論理ノードグループ入力	ECOFAC	エラー	エラー	エラー	シフト 6	エラー	エラー
		CII	エラー	シフト 1	エラー	シフト 2	エラー	エラー
		ユーザ固有フォーマット (マッピングコードなし)	エラー	シフト 1	エラー	シフト 2	エラー	エラー
		ユーザ固有フォーマット (マッピングコードあり)	シフト 3	エラー	エラー	エラー	エラー	エラー

(b)

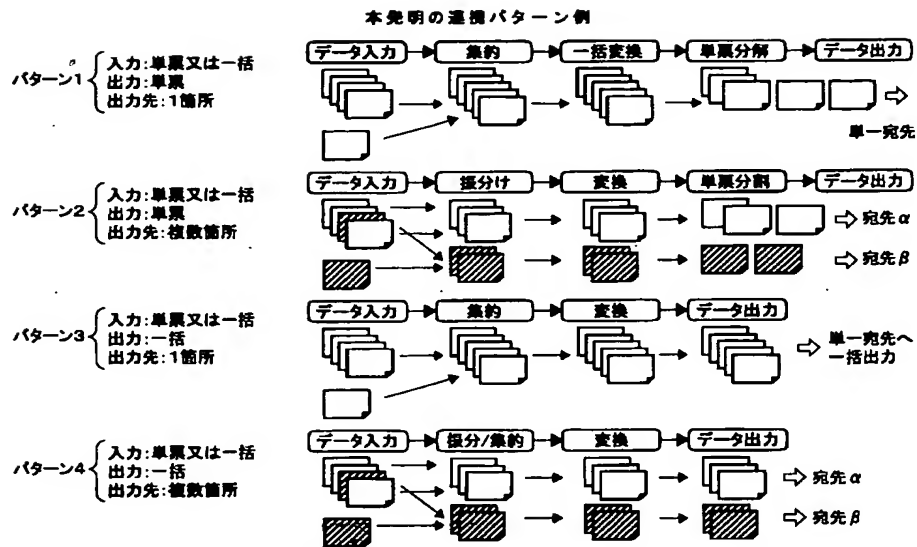
後続業務		出力論理ノードグループ(伝票形式: 単票)		出力論理ノードグループ(伝票形式: 一括)	
		CII	ユーザ固有フォーマット	CII	ユーザ固有フォーマット
先行業務	入力論理ノードグループ入力	ECOFAC	エラー	シフト 6	エラー
		CII	シフト 4	エラー	シフト 5
		ユーザ固有フォーマット (マッピングコードなし)	シフト 4	エラー	シフト 5
		ユーザ固有フォーマット (マッピングコードあり)	エラー	エラー	エラー

【図5】

## 本発明の制御情報と帳票データの例

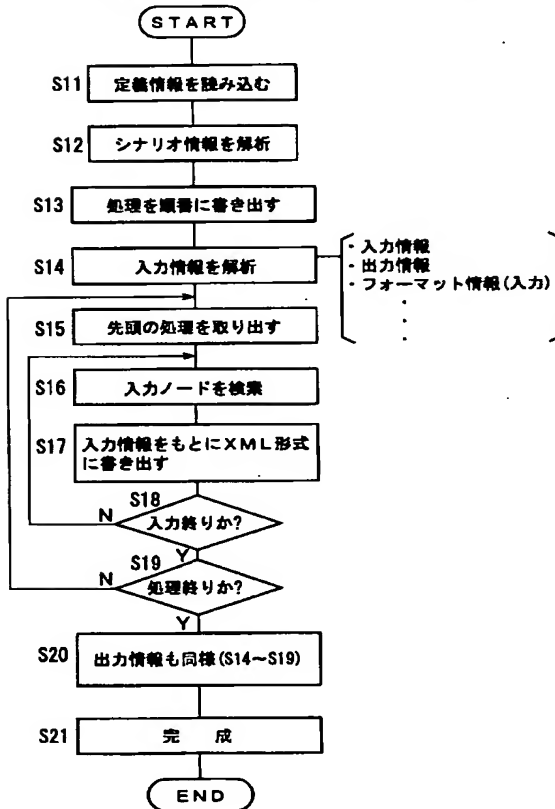


【図6】



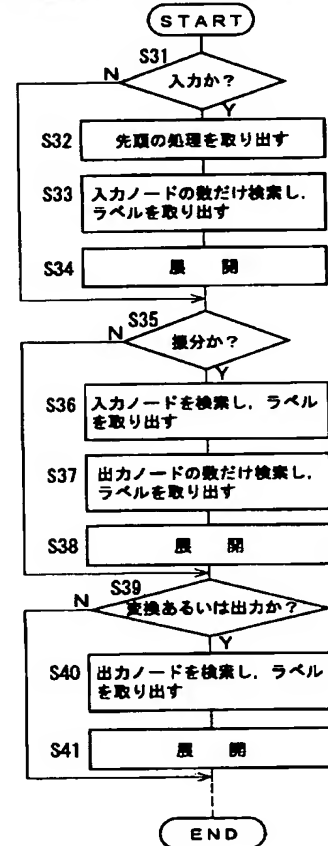
【図7】

本発明の動作説明フローチャート(文書構造化)



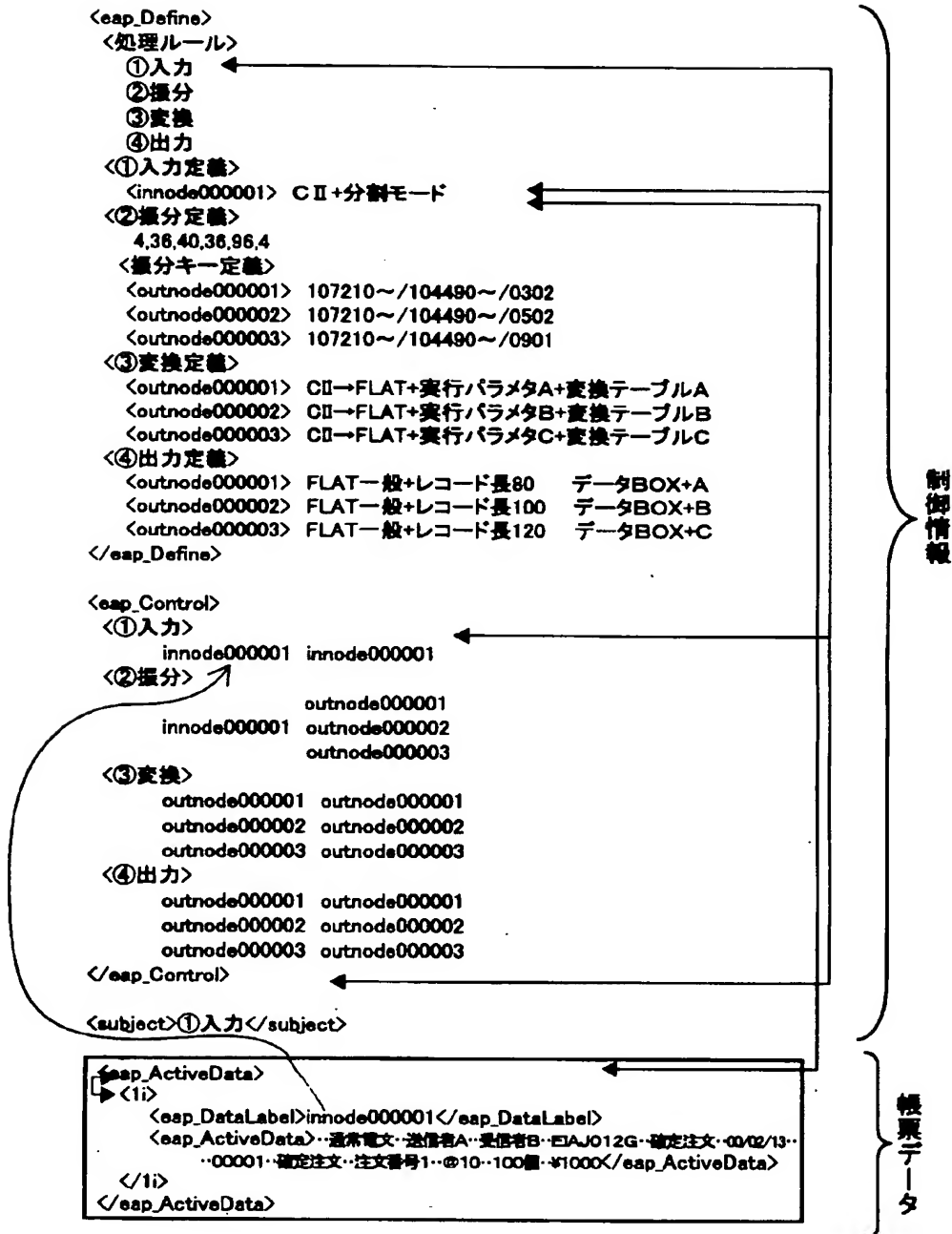
【図8】

本発明の動作説明フローチャート(文書構造化、その2)

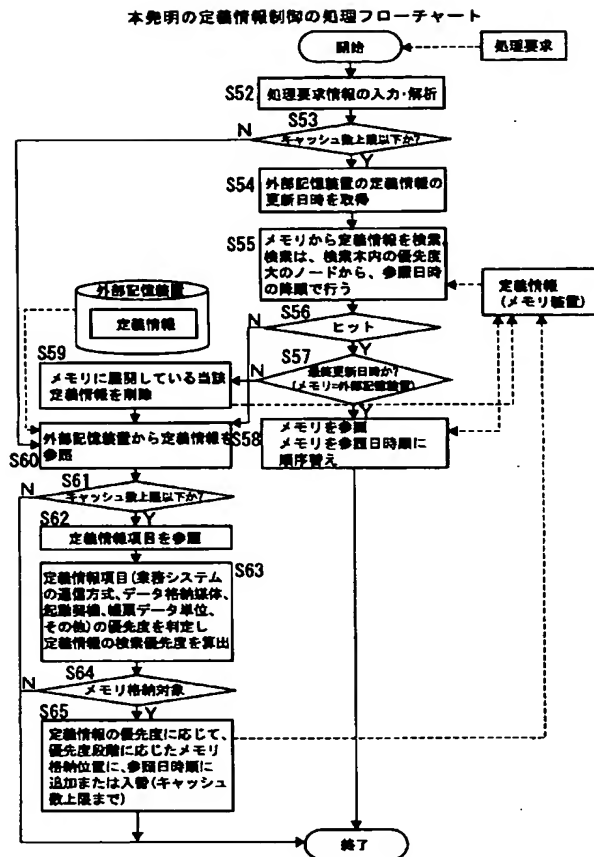


【図9】

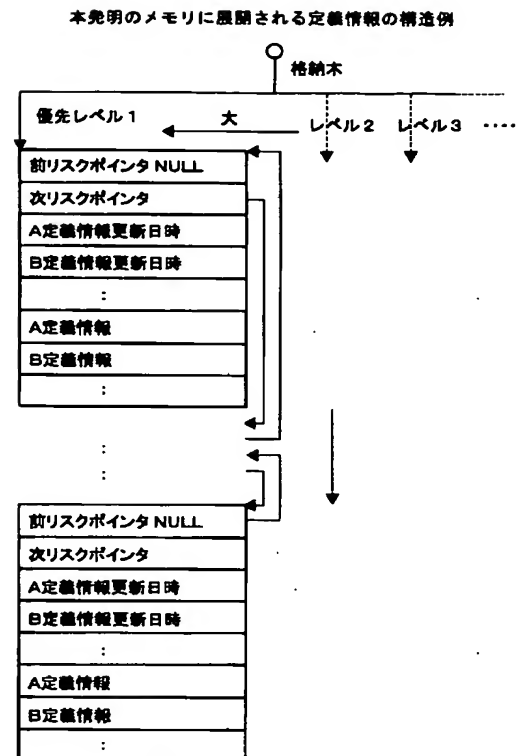
## 本発明の制御情報の文書構造例



【図11】



【図12】



フロントページの続き

(72)発明者 天久 英幸  
 神奈川県横浜市港北区新横浜二丁目15番16  
 号 株式会社富上通ハイパーソフトテクノ  
 ロジ内

Fターム(参考) 5B082 GA02